Facultad de Informática de Madrid

LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

EXAMEN FINAL – JULIO 2014

Apellidos: SOLUCION Nombre:

Ejercicio 1:

Dada la Gramática G = { Σ_T = { a , b }, Σ_N = { S , A }, S, $\mathcal{P}$} con las siguientes producciones:

$$\mathcal{P} \equiv \left| \begin{array}{l} S ::= AAA \\ A ::= AAA \mid bA \mid Ab \mid a \end{array} \right.$$

a) Definir el concepto de gramática ambigua.

b) Comprobar si la gramática G es ambigua o no lo es.

30 minutos

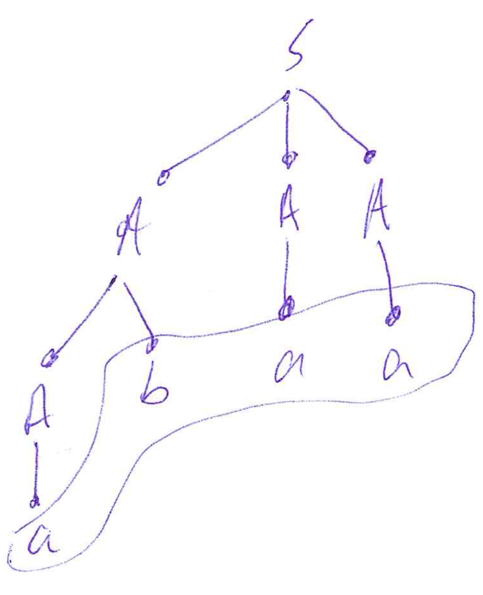
---

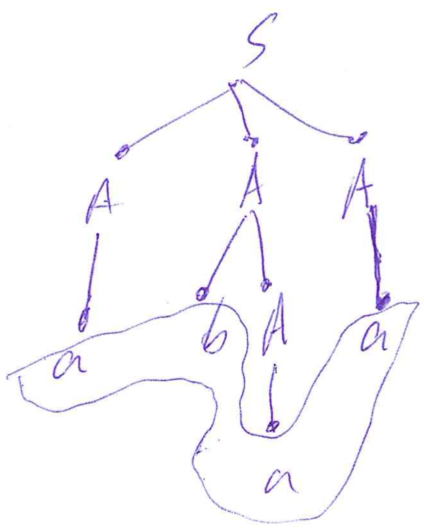
a) Gramática ambigua es la gramática que genera al menos una palabra ambigua
Una palabra es ambigua si al menos admite dos derivaciones por la izquierda diferentes ó admite al menos dos árboles de derivación distintos.

b) x=abaa es palabra ambigua. Por tanto la gramática es ambigua.

$S \rightarrow AAA \rightarrow AbAA \rightarrow abAA \rightarrow abaA \rightarrow abaa$

$S \rightarrow AAA \rightarrow aAA \rightarrow abAA \rightarrow abaA \rightarrow abaa$





Facultad de Informática de Madrid

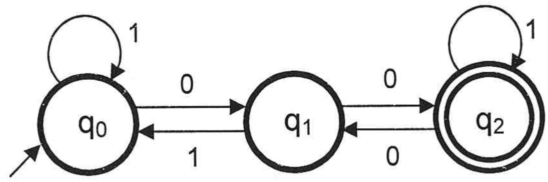
LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

EXAMEN FINAL – JULIO 2014

Apellidos: SOLUCION Nombre:

Ejercicio 2:

Calcular el lenguaje que reconoce el Autómata Finito A mediante ecuaciones características.

A ≡ 

30 minutos

$$\begin{cases} X_0 = 1X_0 + 0X_1 \\ X_1 = 1X_0 + 0X_2 \\ X_2 = 0X_1 + 1X_2 + \lambda \end{cases}$$

$$X_2 = 0X_1 + 1X_2 + \lambda = 1X_2 + (0X_1 + \lambda)$$

$$X_2 = 1^*(0X_1 + \lambda)$$

$$X_1 = 1X_0 + 0X_2 = 1X_0 + 01^*(0X_1 + \lambda) = 1X_0 + 01^*0X_1 + 01^*$$

$$X_1 = (01^*0)^*(1X_0 + 01^*)$$

$$X_0 = 1X_0 + 0X_1$$

$$X_0 = 1X_0 + 0(01^*0)^*(1X_0 + 01^*) = 1X_0 + 0(01^*0)^*1X_0 + 0(01^*0)^*01^*$$

$$X_0 = (1 + 0(01^*0)^*1)^*0(01^*0)^*01^* = L(A)$$

Facultad de Informática de Madrid

LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

EXAMEN FINAL – JULIO 2014

Apellidos: SOLUCION Nombre:

Ejercicio 3:

Sea el Autómata, AP1 = { Σ, Γ, Q, q_0, A_0, *f*, Ø } que acepta por VACIADO DE PILA, con Σ = { a, b }, Γ = { A_0, A }, Q = { q_0, q_1, q_2, q_3, q_4 } y *f* definida mediante los 7 movimientos siguientes:

(1) *f*(q_0 a A_0) = (q_1 AA_0)

(2) *f*(q_1 a A) = (q_2 AA)

(3) *f*(q_2 a A) = (q_3 AA)

(4) *f*(q_3 a A) = (q_3 AA)

(5) *f*(q_3 b A) = (q_4 λ)

(6) *f*(q_4 b A) = (q_4 λ)

(7) *f*(q_4 λ A_0) = (q_4 λ)

a) Construir, utilizando el algoritmo correspondiente, un AP2 que acepte por ESTADOS FINALES el mismo lenguaje que AP1. Siendo AP2 = { Σ, $\Gamma \cup \{A_0'\}$, $Q \cup \{q_0', q_F\}$, q_0', A_0', f', F }, donde F = {q_F}.

b) Comprobad la aceptación de las palabras aabb y aaaabbbb en ambos autómatas a pila.

c) Describe el lenguaje que aceptan AP1 y AP2.

30 minutos

---

a)

1) $f'(q_0'\ \lambda\ A_0') = (q_0, A_0A_0')$ => 1º PASO *f*(q_0' λ A_0') = (q_0, A_0A_0') Accede a la D.I.I. de AP1.

2) $f'(q_0\ a\ A_0) = (q_1\ AA_0)$

3) $f'(q_1\ a\ A) = (q_2\ AA)$

4) $f'(q_2\ a\ A) = (q_3\ AA)$

5) $f'(q_3\ a\ A) = (q_3\ AA)$

6) $f'(q_3\ b\ A) = (q_4\ \lambda)$

7) $f'(q_4\ b\ A) = (q_4\ \lambda)$

8) $f'(q_4\ \lambda\ A_0) = (q_4\ \lambda)$

=> 2º PASO f'(q a A) = *f* (q a A) mismos movimientos f' de AP2 que *f* de AP1.

9) $f'(q_4\ \lambda\ A_0') = (q_F, \lambda)$ => 3º PASO (q_F, λ) $\in f'$(q λ A_0') accede a estado final q_F cuando borra A_0'.

b) **Aceptación AP1:**

Palabra aabb: [q_0 aabb A_0] $\vdash$ [q_1 abb AA_0] $\vdash$ [q_2 bb A_0] **NO ACEPTA**

Palabra aaaabbbb: [q_0 aaaabbbb A_0] $\vdash$ [q_1 aaabbbb AA_0] $\vdash$ [q_2 aabbbb AAA_0] $\vdash$ [q_3 abbbb $AAAA_0$] $\vdash$ [q_3 bbbb $AAAAA_0$] $\vdash$ [q_4 bbb $AAAA_0$] $\vdash$ [q_4 bb AAA_0] $\vdash$ [q_4 b AA_0] $\vdash$ [q_4 λ A_0] $\vdash$ [q_4 λ λ] **ACEPTA**

Aceptación AP2:

Palabra aabb: [q_0' aabb A_0'] $\vdash$ [q_0 aabb A_0A_0'] $\vdash$ [q_1 abb AA_0A_0'] $\vdash$ [q_2 bb A_0A_0'] **NO ACEPTA**

Palabra aaaabbbb: [q_0' aaaabbbb A_0'] $\vdash$ [q_0 aaaabbbb A_0A_0'] $\vdash$ [q_1 aaabbbb AA_0A_0'] $\vdash$ [q_2 aabbbb AAA_0A_0'] $\vdash$ [q_3 abbbb $AAAA_0A_0'$] $\vdash$ [q_3 bbbb $AAAAA_0A_0'$] $\vdash$ [q_4 bbb $AAAA_0A_0'$] $\vdash$ [q_4 bb AAA_0A_0'] $\vdash$ [q_4 b AA_0A_0'] $\vdash$ [q_4 λ A_0A_0'] $\vdash$ [q_4 λ A_0'] $\vdash$ [q_F λ λ] **ACEPTA**

c) El lenguaje que aceptan AP1 y AP2 es: L = { a^nb^n / $n \geq 3$ }

Facultad de Informática de Madrid

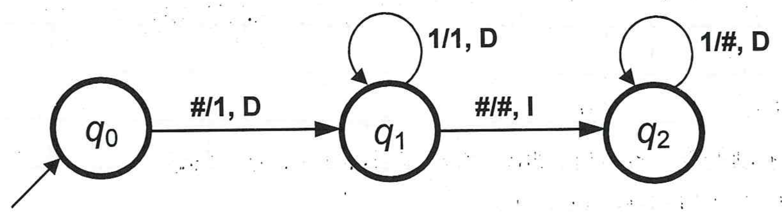
LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

EXAMEN FINAL – JULIO 2014

Apellidos: SOLUCION Nombre:

Ejercicio 4:

Sea la Máquina de Turing M definida según el siguiente grafo:



Y cuya configuración inicial es la siguiente:

x # y

$\uparrow$ q_0

Donde *x* e *y* son dos números enteros positivos codificados en unario.

M inicialmente está en el estado q_0 leyendo el # intermedio.

a) a.1) ¿Qué función aritmética calcula M?

a.2) Escribir (y describir brevemente) el contenido inicial de la cinta de una Máquina de Turing Universal (MTU) cuando simula a la máquina M y ésta recibe como entrada:

1 # 11

(2.5 puntos)

Utilicen la siguiente codificación binaria:

$q_0 \equiv 00$; $q_1 \equiv 01$; $q_2 \equiv 10$

Desplazamiento a la izqda. I ≡ 1; Desplazamiento a la dcha. D ≡ 0

b) Escribir (y describir brevemente) el contenido de la cinta de la MTU después de la ejecución del módulo localizador cuando la MTU está simulando el primer movimiento de M con la entrada del apartado a).

(2.5 puntos)

c) Escribir (y describir brevemente) el contenido de la cinta de la MTU después de la ejecución del módulo transcriptor cuando la MTU está simulando el primer movimiento de M con la entrada del apartado a).

(2.5 puntos)

d) Escribir (y describir brevemente) el contenido de la cinta de la MTU cuando se para después de simular a la máquina M con la entrada del apartado a).

(2.5 puntos)

NOTA: Todos los apartados se responderán en la carilla de atrás.

30 minutos

Facultad de Informática de Madrid
LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD
EXAMEN FINAL – JULIO 2014

Apellidos: Nombre:

Continuación ejercicio 4. RESPUESTAS. SOLUCIONES

Apartado a) Función aritmética: Suma unaria: x+y

...	1	*	1	1	0	#	0	0	0	#	0	0	0	0	1	1	0	#	0	1	1	0	1	1	0	#	

0	1	0	1	0	0	1	#	1	0	1	1	0	0	0	#	#									...

Registro inicial: $\# \underbrace{000}_{q_0} \#$ * en la celda que inicialmente lee M

Hay 4 registros: uno por cada movimiento de M

Apartado b) (es suficiente con escribir sólo la parte de la cinta que cambia)

...						#	A	A	A	#	A	A	A	0	1	1	0	#								

El módulo localizador localiza un registro que comienza por la secuencia de la Referencia inicial (000). El mód. localizador hace 3 bucles de comparación. Se marcan los símbolos de la Ref. inicial y del REG localizado

Apartado c) (es suficiente con escribir sólo la parte de la cinta que cambia)

...						#	0	1	1	#	A	A	A	A	B	B	A	#								

El mód. transcriptor transcribe la información del REG localizado en las celdas de la REF inicial. El último símbolo del REG localizado no cabe en la REF inicial por lo que se memoriza. Cada símbolo transcrito se marca con A o B (si era 0 o 1).

Apartado d)

...	1	1	1	0	*	#	B	0	0	#	A's		B's		#	A's B's	#	#

M se para: $\#111\underset{q_2}{\#}\#$

Por tanto la MTU cuando se para tiene: $1110{*}\#\underset{q_2}{B}00\underset{\#}{\#}$

Todos los registros marcados con A's y B's porque han sido rechazados por el módulo localizador